<!DOCTYPE html>

<html lang="ar" dir="rtl">

<head>

<base target="\_top">

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>G-Assistant</title>

<style>

:root {

--bg-primary: #1f2937; --bg-secondary: #374151; --bg-tertiary: #4b5563;

--text-primary: #f9fafb; --text-secondary: #d1d5db; --text-accent: #60a5fa;

--border-color: #4b5563; --accent-color: #3b82f6; --accent-hover: #2563eb;

--error-color: #ef4444; --success-color: #22c55e;

}

body {

font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

background-color: var(--bg-primary);

color: var(--text-primary);

margin: 0;

padding: 1rem;

display: flex;

flex-direction: column;

height: 100vh;

box-sizing: border-box;

}

.header { text-align: center; margin-bottom: 1rem; }

.header h2 { font-size: 1.5rem; color: var(--text-accent); margin: 0; }

.response-area {

flex-grow: 1;

overflow-y: auto;

background-color: var(--bg-secondary);

border-radius: 8px;

padding: 1rem;

border: 1px solid var(--border-color);

margin-bottom: 1rem;

white-space: pre-wrap;

word-wrap: break-word;

}

.input-area { display: flex; gap: 0.5rem; }

textarea {

flex-grow: 1;

padding: 0.75rem;

background-color: var(--bg-tertiary);

border: 1px solid var(--border-color);

color: var(--text-primary);

border-radius: 8px;

font-size: 1rem;

resize: none;

min-height: 50px;

}

textarea:focus { outline: 2px solid var(--accent-color); }

button {

padding: 0.75rem 1.5rem;

background-color: var(--accent-color);

color: var(--text-primary);

border: none;

border-radius: 8px;

font-weight: bold;

cursor: pointer;

transition: background-color 0.2s;

}

button:hover { background-color: var(--accent-hover); }

button:disabled { background-color: var(--bg-tertiary); cursor: not-allowed; }

.message {

padding: 0.75rem; margin-bottom: 0.5rem; border-radius: 6px;

border-left: 4px solid;

}

.message.info { border-color: var(--accent-color); background-color: var(--bg-secondary); }

.message.success { border-color: var(--success-color); background-color: rgba(34, 197, 94, 0.1); }

.message.error { border-color: var(--error-color); background-color: rgba(239, 68, 68, 0.1); }

table { width: 100%; border-collapse: collapse; margin-top: 1rem; }

th, td { padding: 0.5rem; border: 1px solid var(--border-color); text-align: right; }

th { background-color: var(--bg-tertiary); }

</style>

</head>

<body>

<div class="header">

<h2>✨ G-Assistant</h2>

</div>

<div id="responseArea" class="response-area">

<p>أهلاً بك! كيف يمكنني مساعدتك اليوم؟</p>

</div>

<div class="input-area">

<textarea id="promptInput" placeholder="اكتب طلبك هنا..." rows="2"></textarea>

<button id="sendButton" onclick="sendPrompt()">إرسال</button>

</div>

<script>

const sendButton = document.getElementById('sendButton');

const promptInput = document.getElementById('promptInput');

const responseArea = document.getElementById('responseArea');

function setLoading(isLoading) {

sendButton.disabled = isLoading;

promptInput.disabled = isLoading;

sendButton.textContent = isLoading ? '...' : 'إرسال';

}

function displayResponse(response) {

responseArea.innerHTML = ''; // Clear previous response

const messageDiv = document.createElement('div');

messageDiv.className = `message ${response.type}`;

if (response.type === 'table' && response.data) {

let tableHtml = `<h4>${response.text}</h4><table><thead><tr>`;

response.data.headers.forEach(h => tableHtml += `<th>${h}</th>`);

tableHtml += `</tr></thead><tbody>`;

response.data.rows.forEach(r => {

tableHtml += `<tr>`;

r.forEach(c => tableHtml += `<td>${c}</td>`);

tableHtml += `</tr>`;

});

tableHtml += `</tbody></table>`;

messageDiv.innerHTML = tableHtml;

} else {

messageDiv.textContent = response.text;

}

responseArea.appendChild(messageDiv);

}

function sendPrompt() {

const prompt = promptInput.value.trim();

if (!prompt) return;

setLoading(true);

responseArea.innerHTML = '<p>جاري المعالجة...</p>';

google.script.run

.withSuccessHandler(response => {

setLoading(false);

displayResponse(response);

promptInput.value = '';

})

.withFailureHandler(error => {

setLoading(false);

displayResponse({ type: 'error', text: `حدث خطأ في الخادم: ${error.message}` });

})

.askGemini(prompt, {});

}

promptInput.addEventListener('keydown', (event) => {

if (event.key === 'Enter' && !event.shiftKey) {

event.preventDefault();

sendPrompt();

}

});

</script>

</body>

</html>